41.00 - 208 Reissue

ELECTRONIC HAND HELD LOGIC GAME

BACKGROUND OF THE INVENTION

This invention relates generally to an electronic game and in particular to an electronic logic game wherein a plurality of electric switches and multi-color displays are provided. It is possible, by depressing the switches in a particular manner or pattern, and by observing the resulting colors displayed, to determine the pattern of switches which results in a singular color being indicated on all multi-color displays.

Various logic games are known wherein a plurality of playing pieces of various colors are connected together in a geometric shape, and are manipulated by the player so that pieces of the same color are grouped together. However, such logic games are of mechanical designs and to the inventor's knowledge have never been implemented by "state of the art" electronics, i.e. integrated circuits, etc., which are presently available.

Logic games are generally based on logic problems and can, therefore, be solved using a systematic approach, wherein a player, who is familiar with the logic problem of a particular game, observes the effect caused by a move or a sequence of moves of playing pieces, to determine the next logical steps in the game, and ultimately discover a solution to the problem.

Accordingly, one object of the present invention is to define a logic problem upon which a game may be based, and to provide an electronic device with a field of play whereon a player may discover a solution to the logic problem using the cause/effect characteristics of logic games.

It is another object of this invention to provide an electronic game utilizing electric switching means to control the colors indicated on multi-color displays, and wherein a player must determine the exact combination of switches that results in a singular color being indicated on all multi-color displays.

It is another object of the present invention to provide an electronic game that utilizes a microprocessor to provide a plurality of games by automatically generating random code patterns, and to control the progress of the game.

It is still another object of the present invention to provide an electronic logic game which employs means for varying the level of difficulty of any particular game.

It is still a further object of the present invention to provide an electronic logic game which provides a variety of visual and audible signals to highten the enjoyment of the game.

It is still an other object of the present invention to provide a hand held electronic logic game having a liquid crystal display whereon a plurality of geometric shapes may be depicted in various colors.

SUMMARY OF THE INVENTION

The foregoing and other objects of the invention are accomplished by an electronics logic game which, for demonstration purposes, is graphically represented in FIG. 1 as a geometric square. The section titled "Mathematical Description of the Logic Problem", below, provides definitions and a theoretical description of the logic problem upon which an object of the electronic game herein is based.

Thus the present invention relates to an electronic game comprising means for generating electrical oper-

ating codes, a plurality of electrical switches to control the routing of operating codes within the device, means to route or simulate the routing of operating codes within the device, means to implement a logic boolean function to generate color codes from pairs of operating codes, means to distribute col r codes to multi-color displays, and plurality of multi-color light emitting means to provide multi-color displays.

The present invention defines the logic problem of matching a plurality of objects placed at the left and bottom edges of a square with identical objects placed at its top and right edges, using a plurality of playing pieces, defined as routing squares, to determine the internal routes within the square which interconnect all pairs of objects that belong to a predetermined subset of all possible pairs of said objects.

The present invention also relates to a method of solving the logic problem herein, comprising the definitions of the Routing Square and associated binary switches, designating a color to each predetermined subset of pairs of objects, causing the color associated with each subset to be displayed at multi-color displays according to the position of binary switches, and observing said color displays for different combinations of said switches whereby a combination associated with one subset may be discovered.

In accordance with a preferred embodiment of the invention, there is provided a device having a field of play arranged in an array of multi-color lighted switches on which a player attempts to discover the combination of switch positions which cause a singular color to be displayed on the field of play. The device utilizes a microprocessor programmed to generate random operating code patterns that correspond to objects placed along the edges of a square, simulate the routing of the operating codes from the left and bottom edges to the right and top edges of said square, generate color codes from pairs of operating codes, distribute color codes to multi-color displays, and control the progress of the game. The microprocessor is also programmed to monitor the position of the switches, control the display of multi-color indications, and generate distinct tone sequences representing color melodies and game completion melody. The microprocessor is also programmed to varry the degree of difficulty of each game by randomly rearranging either the switches which control the routing squares, the multi-color displays or both.

In an alternative embodiment, the device comprises a liquid crystal display whereon a plurality of geometric shapes may be depicted and wherein a player attempts to discover a pattern of switch positions that results in a singular geometric shape being depicted at all locations on the liquid crystal display.

In other alternative embodiments, the device comprises an interface module to provide multi-color displays on an external color video monitior.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other more detailed and specific objectives will be disclosed in the course of the following description taken in conjunction with the accompanying drawings wherein:

FIG. 1 is a geometric representation of the preferred embodiment for the RAINBOWX logic game.

FIGS. *2a* and *2b* depict geometric representations of the routing square, indicating the various routes within

the square, for each of the two states of the associated switch.

FIG. 3 is a perspective view f the preferred embodiment f a device according t the invention.

FIG. 4 is a block diagram of the circuit utilized by the present invention.

FIGS. 5 through 12 are logical flow diagrams illustrating the main program functions performed by the microprocessor controlling the operation of the game according to the invention.

FIG. 13 illustrates a flow diagram of the logic steps utilized by the present invention to generate a set of N random numbers.

FIGS. 14 and 15 illustrate a flow diagram of the logic steps utilized by the present invention to generate and assign random operating codes.

FIG. 16 illustrates a flow diagram of the logic steps utilized by the present invention to randomly rearrange switch positions.

FIG. 17 illustrates a flow diagram of the logic steps utilized by the present invention to randomly rearrange display positions.

FIG. 18 indicates legends and explanations of the program variables utilized in the logical flow diagrams of FIGS. 19–22. FIG. 19 is a logical flow diagram illustrating the logic steps utilized by the present invention to determine all pairs of interconnected objects.

FIG. 20 is a logical flow diagram illustrating the logic steps utilized by the present invention to generate color codes at the top and right edges of the square.

FIGS. 21 and 22 illustrate flow diagrams of the logic steps utilized by the present invention to identify all display routes. within the square and to determine the color to be displayed at each multi-color display.

FIGS. 23 and 24 provide proposed operating code and color code assignments, using the EXCLUSIVE OR boolean function for four and eight color games respectively."

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to the drawings where the illustrations are for the purpose of describing the preferred embodiment of the invention and are not intended to limit the invention hereto, FIG. 3 is a front planview of an electronic RAINBOWX device 10 is comprised of a case 12 having a face 14 and carrying an array of individually operable multi-color lighted switches 22 which defining a field of play. In a specific embodiment illustrated in FIG. 3, an arrays of four rows and four columns defines a field of play having sixteen individually operable multi-color lighted switches which may be referred to as 21-1 through 22-16; each row being numbered from left to right and from top to bottom.

A block diagram of the control circuitry for this RAINBOWX device 10 is illustrated in FIG. 4. This control circuitry includes a central processing unit 30 having a control program memory 32 associated therewith, a read only memory (ROM) 32, a random access memory (RAM) 34, a plurality of interface and coding devices 38, 40, 42 and a plurality of memory decoder drivers 36, 44, 48. The interface and coding devices 38, 40, 42 are used as an input interface between the multicolor lighted switches and control push buttons with the central processing unit 30. As such, interface and coding device 38 is associated with level selector switch 18; interface and coding device 40 is associated with sixteen (16) multi-color lighted switches; and interface

and coding device 42 is associated with the new game selector switch. In contrast, the memory decoder device 36 is used as an utput interface between the central processing unit 30 and the multi-color displays. A common address and control bus 52, and a seperate common data bus 50 are used to interconnect the central processing unit 30 with the interface and coding devices 30, 40, 42, the memory decoder drivers 36, 44, 48, the read only memory (ROM) 32, and the random access memory (RAM) 34.

The central processing unit 30 controls the flow of all information throughout the entire system under the direction of the control program. The control program resides in the read only memory (ROM) 32.

A plurality o dry cell batteries 62 are positioned in the housing beneath the switches, these batteries 62 providing power for the central processing unit 30 as well as the multi-color displays 24. An on/off toggle switch 16 is provided to control the operational state of the device and the connection of the internal battery supply 62 to the electric circuitry. A new game selector, push button switch 20, permits the user to terminate the current game and initiate the play of a new one. A level selector rotary switch 18 permits the user to select one of four levels of difficulty playable by the device. A loudspeaker 46 is positioned in the middle portion of the housing and perforations 26 are provided to permit sounds from the loudspeaker 46 to issue from the housing.

With respect to the operation of the device, the logic steps utilized are illustrated in flow diagram form in FIGS. 5 through 12, which interconnect with each other at the places shown in the various figures. Even though specific reference will not be made to this diagram in the following description of the operation of the device, periodic reference to this diagram may prove to be helpfull to the reader hereof.

Referring again to FIG. 4, in order to operate the device, the player moves the off-on switch 16 from the "off" position to the "on" position which causes power to be supplied to all terminals of the device 10 from either a battery 62 or some external power source and which causes a pulse generator 64 to generate a reset pulse. This pulse is applied to the central processing unit 30 and causes the central processing unit 30 to clear any data remaining in the RAM 34 and in the memory decoder drivers 36, 44 over the common data bus 50. The pulse also causes the central processing unit 30 to generate four (4) sets of random numbers. Each of said sets of random numbers comprises four (4) distinct decimal numbers from 1 to 4. and each of said distinct decimal number s corresponds to a location (1 to 4) at an edge of the geometric square described in FIG. 1 such that the first set of random numbers corresponds to the four locations at the left edge of the square, the second set of random numbers corresponds to the four locations at the bottom edge of the square, the third set of random numbers corresponds to the four locations at the top edge of the square and the fourth set of random numbers corresponds to the four locations at the right edge of the square. The central processing unit 30 also assigns the four binary numbers 000, 001, 010 and 011 to the four locations at the left edge of the square such that the binary number 000 is assigned to the location identified by the first decimal number of the first random set, the binary number 001 is assigned to the location identified by the second decimal number of the first random set, etc. . . . Similarly, the four binary number 100, 101, 110

and 111 are assigned to the f ur locati ns at the bottom edge of the square, the f ur binary numbers 000, 001, 010 and 011 are assigned t the f ur locations at the top edge of the square, and the f ur binary numbers 100, 101, 110 and 111 are assigned to the four locations at the right edge of the square. Next, the level selector switch 18 through the interface and coding device 38 accesses the central processing unit 30 over the address and control bus 52 and a signal is transmitted thereto via the data bus 50. The central processing unit 30 identifies the level of difficulty, i.e., the position of the level selector switch 18, and through its control program 32 rearranges switch positions 21-1 through 22-16 and/or multi-color color display positions 24-1 through 24-16, such that if the level selector switch 18 is set to either "2" or "4", the central processing unit 30 generates a set of random numbers which comprises sixteen (16) distinct decimal numbers from 1 to 16, and each of said decimal numbers corresponds to each of the actual positions of switches 22-1 through 22-16, such that if the player activates the switch located at position 22-x, it will appear to the device that the switch located at position 22-y has been activated wherein y is the random decimal number which corresponds to the actual switch position x. Similarly, if the level selector switch 18 is set to either "3" or "4", the central processing unit 30 generates a different set of random numbers which also comprises sixteen (16) distinct decimal numbers from 1 to 16, and each of those decimal numbers corresponds to each of the actual positions of multi-color displays 24-1 through 24-16, such that if the control program 32 determines that the multi-color display located at position 24-z should be activated, the central processing unit 30 will activate the multi-color display located at position 24-w, and it will appear to the player that the display located at position 24-w has been activated wherein w is the random decimal number which corresponds to actual display position z. At any time during the course of a game, the player may change the position of the level selector switch 18, however, only two (2) sets of random numbers are generated by the central processing unit 30 for each single game (one set for apparent switch positions and a second set for apparent display positions). At all times during the course of a single game, the central processing unit 30 stores the current position of the level selector switch 18 in RAM 34, identifies any new position of said switch, and through its control program 32 rearranges or restores the positions of switches 22-1 through 22-16 and/or rearranges or restores the positions of multi-color displays 24-1 through 24-16, as the case may be, and as fully illustrated in flow diagram form in FIG. 6.

To determine the initial status of all switches 22-1 through 22-16, the central processing unit 30 accesses each of said switches over the address and control bus 52 and interface and coding device 40 causing a signal to be transmitted thereto via the data bus 50. The central processing unit 30 identifies the status of the switch, i.e., if the switch is in the "ON" ("1") or "OFF" ("0") position. The central processing unit 30, through its control program 32, identifies the RAM memory address which corresponds to the switch and accesses this memory address over the address control bus 52. The central processing unit 30 then transfers the data on the status of the switch to said RAM memory address over the data bus 50. After the initial status of all switches are stored in RAM 34, the central processing unit 30 through its control program 32 identifies an opcode

receiver "R" for each opcode transmitter "T". As illustrated in the flow diagram of FIG. 19, the control program 32 first determines if transmitter "T" is located at either the left edge or the bottom edge of the square, then it determines the location of the first switch adjacent to said transmitter "T". Starting at this location, the control program 32 traces an internal route within the square by using the status of said first switch to determine the location of the second switch on the route. The status of the second switch is then used to determine the location of the third switch on the route, etc. . . . The foregoing process continues until this internal route terminates at an opcode receiver "R" located at either the top edge or the right edge of the square. The central processing unit 30 through its control program 32 causes the locations of transmitter "T" and associated receiver "R" to be stored in RAM 34.

After the locations of all opcode transmitters and associated opcode receivers are stored in RAM 34, the central processing unit 30, through its control program 32, generates a color code at each opcode receiver. As illustrated in the flow diagram of FIG. 20, the central processing unit 30, through its control program 32, identifies the transmitter associated with the receiver at location "1" by accessing the RAM 34 over the address and control bus 52 causing the identity of said transmitter to be transmitted to the central processing unit 30 via the data bus 50. The central processing unit 30, under the instruction of the control program 32, then accesses the RAM 34 over the address and control bus 52 to obtain the two opcodes assigned to receiver "1" and its associated transmitter. The RAM then forwards said two opcodes over the data bus 50 to the central processing unit 30. To generate the color code at receiver "1", the central processing unit executes the "INCLUSIVE OR" boolean function on the third (left) digit of the opcode assigned to receiver "1" and the third (left) digit of the opcode assigned to the transmitter associated with receiver "1", to compute the third (left) digit of said color code. Similarly, the first and second digits of the color code are computed from the opcodes using the "EXCLUSIVE OR" boolean function. The central processing unit 30 then causes said color code at receiver "1" to be stored in RAM 34. The foregoing processing continues until all eight (8) color codes at the eight (8) opcode receivers are computed and stored in RAM 34.

The central processing unit 30, through its control program 32, then identifies the locations of the multi-color displays connected to each opcode receiver and assigns the color code generated at the receiver to either the top edge or the right edge of the routing square associated with each multi-color display connected to said opcode receiver. As illustrated in the flow diagram of FIG. 21, for each receiver "R", the control program 32 first determines if the receiver "R" is located at either the top edge or the right edge of the square, then it determines the location of the first switch and multi-color display adjacent to said receiver "R". If "R" is located at the top edge of the square, the central processing unit 30, through its control program 32, assigns the color code generated at receiver "R" to the top edge of the routing square associated with the first multi-color display. Alternatively, if "R" is located at the right edge of the square, the central processing unit 30, through its central program 32, assigns the color code generated at the receiver "R" to the top edge of the routing square associated with the first multi-color

display. Starting at this location of first multi-color display, the contr l pr gram 32 traces an internal route within the square by using the status f the first switch to determine the location of the second switch and multi-col r display on the route. The status of the second switch is then used to determine the location of the third switch and multi-color display on the route, etc. . . . The foregoing process continues until this internal route terminates at either the left edge or the bottom edge of the square. While this is occurring, the central processing unit 30 also assigns the color code generated at receiver "R" to either the top edge or the right edge of the routing square associated with each multi-color display on the route. The central processing unit 30, under the instruction of the control program 32, then causes the color codes assigned to either the top edge or the right edge of the routing square associated with each multi-color display on the route to be stored in RAM 34. The foregoing operation is employed to identify all display routes within the square and to assign two color codes to each multi-color display.

The central processing unit 30, through its control program 32, then selects a color code to activate each of the sixteen (16) multi-color displays. As illustrated in the flow diagram of FIG. 22, for the multi-color display associated with the routing square located at row I and column J of the geometric square described in FIG. 1, the control program uses the status of the switch, also located at row I and column J, to determine the color to be forwarded to this multi-color display, such that if the status of said switch is "0", then the color code assigned to the top edge of the routing square is forwarded to the multi-color display, and if the status of said switch is "1", then the color code assigned to the right edge of the routing square is forwarded to the multi-color display. The central processing unit 30 also causes the selected color code to be stored in RAM 34. The foregoing process continues until all sixteen (16) selected color codes are store in RAM.

In order to activate the multi-color displays, the central processing unit 30, through its control program 32, identifies the selected color code addresses in RAM 34, and over the address and control bus 52 accesses said RAM addresses. The RAM 34, in turn, transfers color codes data over the data bus 50 to the memory decoder driver 36 via the central processing unit 30. The memory decoder driver 36, in turn, activates each of the sixteen (16) multi-color displays such that if the first (left) digit of the selected color code equals to "1", then if the second and third digits equal to "00", then the display will indicate "RED"; if the second and third digits equal to "01", then the display will indicate "YELLOW"; if the second and third digits equal to "10", then the display will indicate "GREEN" and if the second and third digits equal to "11", then the display will indicate "BLUE". Alternatively, if said first digit equals to "0", then the display will be "DARK".

After the multi-color displays have been updated in accordance with the initial positions of the switches, the determination is made by the central processing unit in a decision block SAME COLOR? as to whether or not all multi-color displays indicate the same color. If the determination is NO, the central processing unit 30, through its control program, transfers the distinct tone sequences of the ready beep to the memory decoder 44 over the data bus 50. The memory decoder and associated audio control circuits 44, in turn, causes said tone sequences to be generated through the loud speaker 46.

The device 10 is now ready for the player to activate one or more switches in order t solve the puzzle.

If the player activates any f the switches 22-1 through 22-16, the interface and coding device 40 accesses the central processing unit 30 ver the address and control bus 52 and a signal is transmitted theret via the data bus 50. The central processing unit identifies the position of the activated switch 22, and the status of said switch, i.e., if the switch is in the "ON" ("1") or "OFF" ("0") position. The central processing unit 30, through its control program 32, then identifies the associated RAM memory address over the address control bus 52, and causes the data on the status of said switch to be transferred to the RAM memory address over the data bus 50. The central processing unit 30, under the instruction of the control program 32, also scans all remaining switches 22, as well as the level selector switch, and causes the status of said switches to be transferred to the RAM 34 over the data bus 50.

After the detection of any changes in switch positions, the central processing unit 30, through its control program 32, transfers a signal to the memory decoder 44, over the data bus 50, causing said memory decoder and associated audio control circuits 44, to generate a high pitch beep tone through the loud speaker 46. The logic control then proceeds to perform the functions of identifying an opcode receiver "R" for opcode transmitter "T", generating a color code at each opcode receiver, identifying the locations of the multi-color displays connected to each opcode receiver, assigning two color codes to each routing square, selecting a color code to update each of the sixteen (16) multi-color displays and transfering color codes data, over the data bus 50, to the memory decoder drivers 36 to update said multi-color displays.

After these functions have been performed, the determination is again made by the central processing unit 30 in the decision block SAME COLOR? as to whether or not all multi-color displays indicate the same color. If the determination is still NO, the player may continue to activate the switches causing the central processing unit, under the instruction of the control program, to repeat the foregoing operation.

Upon the determination that the same color is indicated at all sixteen (16) displays, the central processing unit 30, through its control program 32, identifies the color being displayed, selects a melody from a plurality of melodies associated with said color and stored in the control program memory 32, and sets the display code to the color code of the color being displayed. The central processing unit 30 also accesses the memory decoder driver 48 over the address and control bus 52 and transmits a signal over the data bus to activate the flashing control circuit 56 causing all multi-color displays to flash their indications. The central processing unit 30, through its internal timer or oscillator circuit then initializes a flashing timer to control the flashing duration of the multi-color displays.

Upon the expiration of the flashing time, the central processing unit 30 deactivates the flashing control circuits 56, and initializes its internal tone generator with the distinct tone sequences of the selected melody. The central processing unit, under the instruction of the control program, transfers said distinct tone sequences to the memory decoder 44 over the data bus 50. The memory decoder and associated audio control circuits 44, in turn, causes the distinct tone sequences of the selected melody to be generated through the loud

speaker 46. While this is occuring, the central processing unit 30 also generates a sequence of random singular color displays, which are synchronized with the tones generated through the loud speaker 46, as fully illustrated in flow diagram form in FIG. 9. The central processing unit, through its control program, first determines the type of the next tone to be generated, then it searches its control program memory 32, to determine the number of multi-color displays associated with that tone. The random locations of said multi-color displays are then transmitted by the central processing unit and the color codes associated with these displays are set to the display code. The color codes for all remaining multi-color display locations are set to "000". The central processing unit then waits for an internal signal before updating the multi-color displays. The foregoing operation continues for each tone generated until a determination is made, by the central processing unit 30, in the decision block DONE MELODY? that the tone sequences of the selected melody have been completed.

Upon the completion of the tone sequences of the selected melody, the logic control flow disables the tone generator then proceeds to a decision block where the determination is made whether or not all color flags have been set to a "1". If the determination is NO, the control path proceeds through the marker D of FIG. 9 to the reference marker D of FIG. 5, so that the player may continue solving the remaining color(s) of the game. If the determination is YES, i.e., all four (4) colors have been solved, the central processing unit 30, through its control program 32, selects and end of game melody from a plurality of melodies stored in the control program memory 32, accesses the memory decoder driver 48 over the address control bus 52 and transmits a signal over the data bus to activate the flashing control circuits 56 causing all multi-color displays to flash their indications. The central processing unit 30, through its internal timer or oscillator circuit, then initializes a flashing timer to control the flashing duration of the multi-color displays. Within said flashing duration, the central processing unit 30 selects one of the four (4) color codes "100", "101", "110" and 111", at random, and assigns it to all sixteen (16) multi-color displays. The central processing unit then waits for an internal signal before updating the multi-color displays. The foregoing process of randomly varying the color of the multi-color flashing displays continues until the expiration of the flashing timer.

Upon the expiration of the flashing time, the central processing unit deactivates the flashing control circuits 56, and initializes the internal tone generator with the distinct tone sequences of the selected end of game melody. The central processing unit 30, under the instruction of the control program 322, transfers said distinct tone sequences to the memory decoder 44 over the data bus 50. The memory decoder and associated audio control circuits 44, in turn, causes the distinct tone sequences of the selected melody to be generated through the loud speaker 46. While this is occurring, the central processing unit 30 also generates a sequence of random multi-color displays which are synchronized with the tones generated through the loud speaker 46 as fully illustrated in flow diagram form in FIGS. 10 and 11.

Upon the completion of the tone sequences of the selected melody, the central processing unit disables the tone generator, sets all color flags to "0", sets the color codes of all sixteen (16) multi-color displays to "000",

and sets all multicol r displays to "DARK". The logic fl w then proceeds through the marker J of FIG. 11 and the reference marker J of FIG. 12 to the decision block NEW GAME? to determine whether or n t the new game switch 20 has been activated. If the player activates the new game switch 20, the interface coding device 42 accesses the central processing unit over the address and control bus 52 and a signal is transmitted thereto via the data bus 50 causing the new game flag to be set to "1". The central processing unit then causes the pule generator 64 to generate a reset pulse which, when applied to the central processing unit, causes the logic control flow to proceed to the reference marker B of FIG. 5. The reset pulse also causes the central processing unit to clear all data in RAM 34 and in the memory decoder driver 36, 44, 48 over the common data bus 50. The central processing unit also resets all flags and program variables. The logic control then proceeds to generate four (4) new sets of random operating codes and to repeat the functions illustrated in FIG. 5 through FIG. 12.

At any time during the progress of a game, the player may terminate the current game and initiate a new one by two consecutive activations of the new game switch 20. Upon the first activation of the new game switch 20, the central processing unit 30, interrupts its current processing and initializes a timer which establishes a time period within which the player must reactivates the new game switch 20 in order to initiate a new game. If the player fails to reactivate the new game switch within the established time, the logic control flow returns to the point where it was interrupted to continue the current game. Upon the second activation of the new game switch 20, within the established time, the central processing unit 30 causes the pule generator 64 to generate a reset pulse and the logic control flow then proceeds to the reference marker B of FIG. 5 to initiate a new game.

As will be understood by those skilled in the art, many different programs may be utilized to implement the flow charts disclosed in FIG. 5 through FIG. 22. Obviously these programs will vary from one another in some degree. However, it is well within the skill of the computer programmer to provide particular programs for implementing each of the steps of the flow charts disclosed herein. It is also to be understood that the foregoing detailed description has been given for clearness of understanding only and is intended to be exemplary of the invention while not limiting the invention to the exact embodiment shown. Obviously certain modifications, variations and improvements will occur to those skilled in the art upon reading the foregoing. It is therefore to be understood that all such modifications, variations and improvements have been deleted herein for the sake of conciseness and readability but are properly within the scope and spirit of the following claims.

MATHEMATICAL DESCRIPTION OF THE LOGIC PROBLEM

A RAINBOWX LOGIC PROBLEM

Let the logic game herein be represented by a geometric square, and let the surface of the square be subdivided into N^2 multi-color sub-squares, where N den tes the number of colors which may be displayed on any sub-square.

Definition Of Operating And Color Codes

Let D denotes a binary operating code of length n, where $n = \ln N + 1$.

Then

D is a set of all possible values of a binary code of length n.

d_i; $i = 1, \ldots, 2N$, is the ith code of D.

Let

$m_{i,j}$, $i,j = 1, \ldots, 2N$, denotes the pair (d_i, d_j).

M be a set of all possible pairs, $m_{i,j}$, of the operating binary code D.

C denotes a binary color code of length n.

Then

C is a set of all possible values of a binary code of length n.

C_k; $k = 1, \ldots, 2N$, is the kth code of C.

Let M_k be a subset of M, of all pairs (d_i, d_j) which satisfy; $B(d_i, d_j) = C_k$, where B is an appropriate boolean function.

Then the color assignment on the surface of the square is defined as follows:

(i) The nth digit of c_k is used to turn a display "ON" and "OFF".

(ii) The first $(n-1)$ digits of c_k are used to select one out of N colors that may be displayed on the square.

The color assignment for the EXCLUSIVE OR boolean function, and for $N=4$ & $N=8$, are shown for FIGS. 23 and 24 respectively.

Definition Of Routing Square

The Routing Square, $S_{i,j}$, shown in FIG. 2, is defined as a quad routing device which is activated by a two-position (binary) switch, $W_{i,j}$. A total of N^{22} Routing Squares are provided in the logic game herein, and are arranged in a two-dimensional geometric layout. The Routing Square, $S_{i,j}$, is then described as follows:

Let

$S_{i,j}$ denotes routing square (i, j).

$W_{i,j}$ denotes binary switch (i, j).

$t_{i,j}$ denotes the TOP edge of $S_{i,j}$.

$l_{i,j}$ denotes the LEFT edge of $S_{i,j}$.

$r_{i,j}$ denotes the RIGHT edge of $S_{i,j}$.

$b_{i,j}$ denotes the BOTTOM edge of $S_{i,j}$.

Two nodes are connected to each edge of the square, a transmitting node (X), and a receiving node (V). The Routing Square functions as follows:

If

$W_{i,j} = $ "1", then:

$b_{i,j}(X)$ CONNECTS TO $t_{i,j}(V)$.

$l_{i,j}(X)$ CONNECTS TO $r_{i,j}(V)$.

$r_{i,j}(X)$ CONNECTS TO $b_{i,j}(V)$.

$t_{i,j}(X)$ CONNECTS TO $l_{i,j}(V)$.

If

$W_{i,j} = $ "O", then:

$b_{i,j}(X)$ CONNECTS TO $r_{i,j}(V)$.

$l_{i,j}(X)$ CONNECTS TO $t_{i,j}(V)$.

$r_{i,j}(X)$ CONNECTS TO $l_{i,j}(V)$.

$t_{i,j}(X)$ CONNECTS TO $b_{i,j}(V)$.

Definition Of A Rainbowx Logic Game

Having defined the operating & color codes, and the Routing Square, the logic game herein is described as f llows:

As stated, the logic game is represented by a geometric square subdivided into N^2 multi-color sub-squares.

Let

T denotes the TOP edge of the Square.
R denotes the RIGHT edge of the Square.
L denotes the LEFT edge of the Square.
B denotes the BOTTOM edge f the Square.

Then for a dimension "N", each edge is divided int "N" sectors as follows:

Let

$t_{1,j}$; $j=1, \ldots, N$, denote TOP sectors.
$r_{i,N}$; $i=1, \ldots, N$, denote RIGHT sectors.
$l_{i,1}$; $i=1, \ldots, N$, denote LEFT sectors.
$b_{N,j}$; $j=1, \ldots, N$, denote BOTTOM sectors.
X_i; $i=1, \ldots, 2N$, denote Operating Code transmitters.
CG_j; $j=1, \ldots, 2N$, denote Color Code generators.
$CD_{i,j}$; $i,j=1, \ldots, N$, denote Color Code decoders.

The Operating Code transmitters, X_i, are connected to the left and bottom edges of the Square, and the Color Code generators, CG_j, are connected to the top and right edges of the Square, as follows:

X_i; $i=1, \ldots, N$, are connected to $l_{i,1}(X)$; $i=1, \ldots, N$.
X_i; $i=N+1, \ldots, 2N$, are connected to $b_{N,m}(X)$; $j=1, \ldots, N$.
CG_j; $j=1, \ldots, N$, are connected to $t_{1,j}(V)$; $j=1, \ldots, N$.
CG_j; $j=N+1, \ldots, 2N$, are connected to $r_{i,N}(V)$; $i=1, \ldots, N$.

The Color Chart decoders, $CD_{i,j}$, are connected to $S_{i,j}$ as follows:

For i, $j=1, \ldots, N$:
If W_{ij}="1" then CD_{ij} is connected to $t_{ij}(X)$.
If $W_{i,j}$="0" then $CD_{i,j}$ is connected to $r_{i,j}(X)$.

Having described the logic game herein, the logic problem is defined as follows:

1. For EACH game, assign the Operating Codes, d_i & d_j; $i,j=1, \ldots, 2N$, to X_i; $i=1, \ldots, 2N$, and CG_j; $j=1, \ldots, 2N$, as follows:
   d_i; $i=1, \ldots, N$, are RANDOMLY assigned to X_i; $i=i, \ldots, N$.
   d_j; $j=1, \ldots, N$, are RANDOMLY assigned to CG_j; $j=1, \ldots, N$.

   Similarly,
   d_i; $i=N+1, \ldots, 2N$, are RANDOMLY assigned to X_i; $i=N+1, \ldots, 2N$.
   d_j; $j=N+1, \ldots, 2N$, are RANDOMLY assigned to CG_j; $j=N+1, \ldots, 2N$.
2. The Operating Codes, d_i ($i=1, \ldots, 2N$), are then transmitted from X_i to CG_j ($i,j=1, \ldots, 2N$), via the Routing Squares. The actual route for each code, d_i, is dependent on the positions of the binary switches, $W_{i,j}$ ($i,j=1, \ldots, N$).
3. When the Operating Codes, d_i ($i=1, \ldots, 2N$), are received by the Color Code generators, CG_j ($j=1, \ldots, 2N$), they are matched with Operating Codes, d_j ($j=1, \ldots, 2N$), which were assigned to CG_j, ($j=1, \ldots, 2N$), and the operating Code pairs, $m_{i,j}$ ($i,j=1, \ldots, 2N$), are then determined.
4. The Color Codes, C_j ($j=1, \ldots, 2N$), are then generated, by the Boolean Function "B", from $m_{i,j}$ ($i,j=1, \ldots, 2N$).
5. The Color Codes, C_j ($j=1, \ldots, 2N$), are transmitted from CG_j ($j=1, \ldots, 2N$) and received by Color Code decoders, $CD_{i,j}$ ($i,j=1, \ldots, N$), via the Routing Squares, where they are decoded and displayed on the multi-color sub-squares. The actual color displayed at each sub-square is dependent on the position of the binary switches, $W_{i,j}$ ($i,j=1, \ldots, N$).
6. The object of the logic game is for the player to continue to manipulate the binary switches, until all the

Operating Code pairs generated belong to the same subset M_k. At such time, all the multi-col r sub-squares will display the color corresponding to the Col r Code, c_k.

7. By changing the positions of the binary switches, the player can continue to play the game until a different color is displayed on all sub-squares. A total of N colors can be displayed in each game.

8. For a new game, change the assignments of d_i and d_j $(i,j=1, \ldots, 2N)$ to X_i and CG_j $(i,j=1, \ldots, 2N)$.